

AD-A083 734

MARYLAND UNIV COLLEGE PARK COMPUTER SCIENCE CENTER

F/G 12/1

THE EFFECT OF THE FUTURE IN WORK DISTRIBUTION.(U)

FEB 80 G RICART, A K AGRAWALA

AFOSR-78-3654

UNCLASSIFIED

CSC-TR-868

AFOSR-TR-80-0310

NL

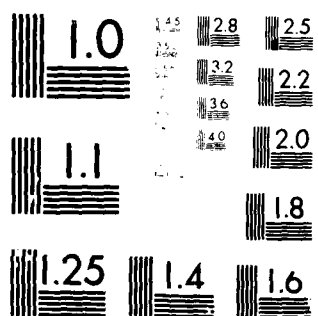
1 of 1
AD
A083734

END

DATE

FILED

DTIC

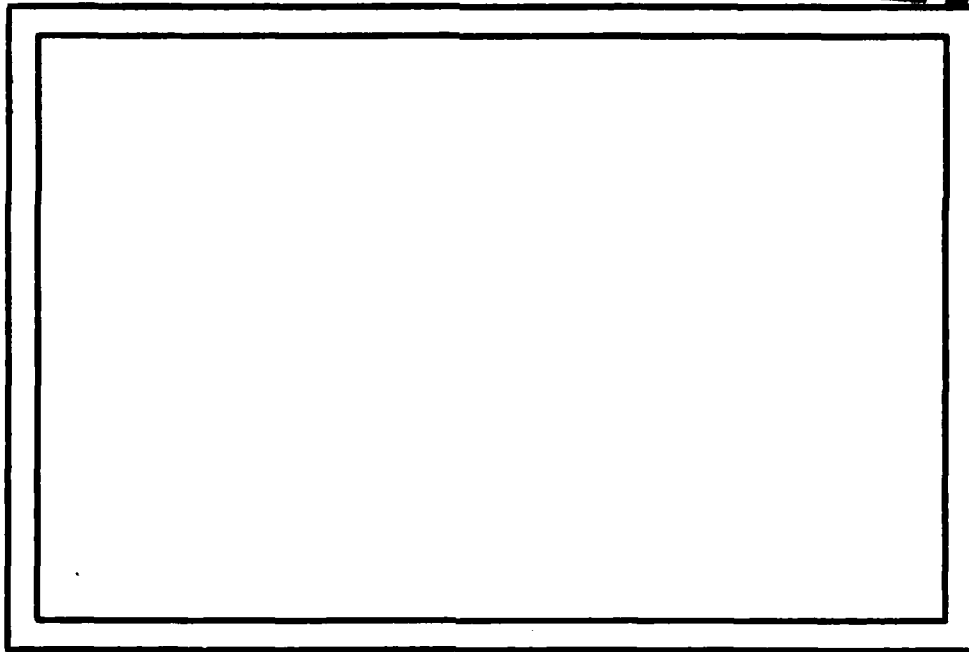


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

12

LEVEL

ADA083734



DTIC
ELECTE
APR 29 1980
A

UNIVERSITY OF MARYLAND
COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

20742

FILE COPY

80 4 21 092

Approved for public release;
distribution unlimited.

Technical Report TR-868
AFOSR78-3654

February 1980

THE EFFECT OF THE FUTURE
IN WORK DISTRIBUTION

Glenn Ricart
Ashok K. Agrawala

DTIC
ELECTE
APR 29 1980
A

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF FINAL REVIEW TO LSC
This technical report has been reviewed and is
approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

This research was supported in part by the Air Force Office of
Scientific Research under grant AFOSR 78-3654. Assistance
was also provided through the National Institutes of Health.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) AFOSR-TR-80-0310	2. GOVT ACCESSION NO. AD-A083734	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) THE EFFECT OF THE FUTURE IN WORK DISTRIBUTION	5. TYPE OF REPORT & PERIOD COVERED Interim	
7. AUTHOR(s) (10) Glenn Ricart Ashok K. Agrawala	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Maryland Computer Science Center College Park, MD 20742	8. CONTRACT OR GRANT NUMBER(s) (15) AFOSR-78-3654	
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NM Bolling AFB, Washington, DC 20332	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) 61102F (17) 2304/A2	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 29/ 403	12. REPORT DATE (11) Feb 80	
	13. NUMBER OF PAGES 25	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE		
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. (14) CSC-TR-869		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) (9) Technical rept.		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A controller is considered which routes arrivals among several servers of different speeds. A decision which sends work to the server which will complete it soonest does not optimize the average completion time (mean flow time) because it doesn't take into account the impact of the decision on future arrivals. This impact on future arrivals, the "future effect", can be significant at high arrival rates. An estimate of the size of the future effect is derived and controllers which take it into account in routing decisions can reduce the average completion time to near optimum. The effect is most pronounced when the		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

403028

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Abstract cont.

Service requirements for arrivals are nearly constant, server speeds are markedly different, and the arrival rate is close to the system's capacity. A controller considering the future effect will more heavily weigh a potential server's backlog than the arrival's service time when making a routing decision.

UNCLASSIFIED

Abstract

A controller is considered which routes arrivals among several servers of different speeds. A decision which sends work to the server which will complete it soonest does not optimize the average completion time (mean flow time) because it doesn't take into account the impact of the decision on future arrivals. This impact on future arrivals, the "future effect", can be significant at high arrival rates. An estimate of the size of the future effect is derived and controllers which take it into account in routing decisions can reduce the average completion time to near optimum. The effect is most pronounced when the service requirements for arrivals are nearly constant, server speeds are markedly different, and the arrival rate is close to the system's capacity. A controller considering the future effect will more heavily weight a potential server's backlog than the arrival's service time when making a routing decision.

Accession For	
1015	✓
1016	
1017	
1018	
1019	
1020	
1021	
1022	
1023	
1024	
1025	
1026	
1027	
1028	
1029	
1030	
1031	
1032	
1033	
1034	
1035	
1036	
1037	
1038	
1039	
1040	
1041	
1042	
1043	
1044	
1045	
1046	
1047	
1048	
1049	
1050	
1051	
1052	
1053	
1054	
1055	
1056	
1057	
1058	
1059	
1060	
1061	
1062	
1063	
1064	
1065	
1066	
1067	
1068	
1069	
1070	
1071	
1072	
1073	
1074	
1075	
1076	
1077	
1078	
1079	
1080	
1081	
1082	
1083	
1084	
1085	
1086	
1087	
1088	
1089	
1090	
1091	
1092	
1093	
1094	
1095	
1096	
1097	
1098	
1099	
1100	
1101	
1102	
1103	
1104	
1105	
1106	
1107	
1108	
1109	
1110	
1111	
1112	
1113	
1114	
1115	
1116	
1117	
1118	
1119	
1120	
1121	
1122	
1123	
1124	
1125	
1126	
1127	
1128	
1129	
1130	
1131	
1132	
1133	
1134	
1135	
1136	
1137	
1138	
1139	
1140	
1141	
1142	
1143	
1144	
1145	
1146	
1147	
1148	
1149	
1150	
1151	
1152	
1153	
1154	
1155	
1156	
1157	
1158	
1159	
1160	
1161	
1162	
1163	
1164	
1165	
1166	
1167	
1168	
1169	
1170	
1171	
1172	
1173	
1174	
1175	
1176	
1177	
1178	
1179	
1180	
1181	
1182	
1183	
1184	
1185	
1186	
1187	
1188	
1189	
1190	
1191	
1192	
1193	
1194	
1195	
1196	
1197	
1198	
1199	
1200	
1201	
1202	
1203	
1204	
1205	
1206	
1207	
1208	
1209	
1210	
1211	
1212	
1213	
1214	
1215	
1216	
1217	
1218	
1219	
1220	
1221	
1222	
1223	
1224	
1225	
1226	
1227	
1228	
1229	
1230	
1231	
1232	
1233	
1234	
1235	
1236	
1237	
1238	
1239	
1240	
1241	
1242	
1243	
1244	
1245	
1246	
1247	
1248	
1249	
1250	
1251	
1252	
1253	
1254	
1255	
1256	
1257	
1258	
1259	
1260	
1261	
1262	
1263	
1264	
1265	
1266	
1267	
1268	
1269	
1270	
1271	
1272	
1273	
1274	
1275	
1276	
1277	
1278	
1279	
1280	
1281	
1282	
1283	
1284	
1285	
1286	
1287	
1288	
1289	
1290	
1291	
1292	
1293	
1294	
1295	
1296	
1297	
1298	
1299	
1300	
1301	
1302	
1303	
1304	
1305	
1306	
1307	
1308	
1309	
1310	
1311	
1312	
1313	
1314	
1315	
1316	
1317	
1318	
1319	
1320	
1321	
1322	
1323	
1324	
1325	
1326	
1327	
1328	
1329	
1330	
1331	
1332	
1333	
1334	
1335	
1336	
1337	
1338	
1339	
1340	
1341	
1342	
1343	
1344	
1345	
1346	
1347	
1348	
1349	
1350	
1351	
1352	
1353	
1354	
1355	
1356	
1357	
1358	
1359	
1360	
1361	
1362	
1363	
1364	
1365	
1366	
1367	
1368	
1369	
1370	
1371	
1372	
1373	
1374	
1375	
1376	
1377	
1378	
1379	
1380	
1381	
1382	
1383	
1384	
1385	
1386	
1387	
1388	
1389	
1390	
1391	
1392	
1393	
1394	
1395	
1396	
1397	
1398	
1399	
1400	
1401	
1402	
1403	
1404	
1405	
1406	
1407	
1408	
1409	
1410	
1411	
1412	
1413	
1414	
1415	
1416	
1417	
1418	
1419	
1420	
1421	
1422	
1423	
1424	
1425	
1426	
1427	
1428	
1429	
1430	
1431	
1432	
1433	
1434	
1435	
1436	
1437	
1438	
1439	
1440	
1441	
1442	
1443	
1444	
1445	
1446	
1447	
1448	
1449	
1450	
1451	
1452	
1453	
1454	
1455	
1456	
1457	
1458	
1459	
1460	
1461	
1462	
1463	
1464	
1465	
1466	
1467	
1468	
1469	
1470	
1471	
1472	
1473	
1474	
1475	
1476	
1477	
1478	
1479	
1480	
1481	
1482	
1483	
1484	
1485	
1486	
1487	
1488	
1489	
1490	
1491	
1492	
1493	
1494	
1495	
1496	
1497	
1498	
1499	
1500	
1501	
1502	
1503	
1504	
1505	
1506	
1507	
1508	
1509	
1510	
1511	
1512	
1513	
1514	
1515	
1516	
1517	
1518	
1519	
1520	
1521	
1522	
1523	
1524	
1525	
1526	
1527	
1528	
1529	
1530	
1531	
1532	
1533	
1534	
1535	
1536	
1537	
1538	
1539	
1540	
1541	
1542	
1543	
1544	
1545	
1546	
1547	
1548	
1549	
1550	
1551	
1552	
1553	
1554	
1555	
1556	
1557	
1558	
1559	
1560	
1561	
1562	
1563	
1564	
1565	
1566	
1567	
1568	
1569	
1570	
1571	
1572	
1573	
1574	
1575	
1576	
1577	
1578	
1579	
1580	
1581	
1582	
1583	
1584	
1585	
1586	
1587	
1588	
1589	
1590	
1591	
1592	
1593	
1594	
1595	
1596	
1597	
1598	
1599	
1600	
1601	
1602	
1603	
1604	
1605	
1606	
1607	
1608	
1609	
1610	
1611	
1612	
1613	
1614	
1615	
1616	
1617	
1618	
1619	
1620	
1621	
1622	
1623	
1624	
1625	
1626	
1627	
1628	
1629	
1630	
1631	
1632	
1633	
1634	
1635	
1636	
1637	
1638	
1639	
1640	
1641	
1642	
1643	
1644	
1645	
1646	
1647	
1648	
1649	
1650	
1651	
1652	
1653	
1654	
1655	
1656	
1657	
1658	
1659	
1660	
1661	
1662	
1663	
1664	
1665	
1666	
1667	
1668	
1669	
1670	
1671	
1672	
1673	
1674	
1675	
1676	
1677	
1678	
1679	
1680	
1681	
1682	
1683	
1684	
1685	
1686	
1687	
1688	
1689	
1690	
1691	
1692	
1693	
1694	
1695	
1696	
1697	
1698	
1699	
1700	
1701	
1702	
1703	
1704	
1705	
1706	
1707	
1708	
1709	
1710	
1711	
1712	
1713	
1714	
1715	
1716	
1717	
1718	
1719	
1720	
1721	
1722	
1723	
1724	
1725	

TABLE OF CONTENTS

1	Introduction	1
2	Distributing Work with Varying Server Speeds	3
3	Task Completion Time	3
4	The Future Effect	4
5	Impact of the Future Effect	7
6	Controllers Considering the Future Effect	11
7	Evaluation	11
	7.1 Method	11
	7.2 Results	13
	7.3 Discussion	14
8	Limitations	16
	8.1 Equal Speed Servers	16
	8.2 Varying Service Times	16
9	Weighting Unfinished Work	19
10	Conclusion	20
	References	22

1. Introduction

The functional distribution of components in computing networks brings with it the problem of distributing work among interchangeable servers. In this paper, a centralized controller (C) is imagined which distributes work among multiple servers N_1 .

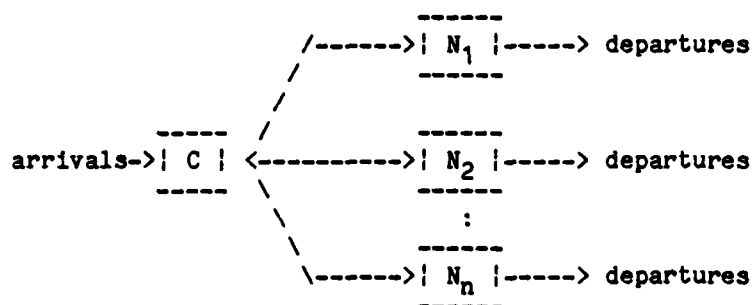


Figure 1.

Each arrival must be immediately dispatched by C to one of the servers; there it may be queued if the server is not free. The servers are work-conserving and operate at fixed speeds.

The goal of C is to distribute work in such a way that the average time to complete service (mean flow time) is minimized. This is equivalent to minimizing the total time in system for the arriving tasks.

The nature of the problem varies with the amount of knowledge that C is presumed to have available for making decisions. For example, if there is no feedback from the servers, the controller C must make its decisions based only upon its knowledge of arrival instants and its own memory of past routing[1].

In this research we permit C to be cognizant of all knowledge in the system except for future arrival instants. In particular, C knows the number of servers, their speeds, their current backlogs, its past routing decisions, and the mean arrival rate. These assumptions correspond to a realistic system with a very good information gathering system.

For the purposes of study, arrivals are generated by a Poisson process. The service times for the arrivals may be constant or variable. Three servers are utilized; to show the effect of varying server speeds, they are given speed ratios of 4:2:1. The mean service time of a task is normalized to 100.0 on the speed 1 server. (The speed 4 server processes the average task in 25.0.)

While this problem is similar to the Multiple Producer / Multiple Consumer Problem [8], it is simplified by assuming that there are no communications delays between the producers and consumers to impede information flow.

The goal of minimizing the average completion time (mean flow time) in a system with servers of different speeds has been previously considered in the case that all work is available at time 0 [2] [7]. An optimal algorithm to minimize mean flow time [4] when the servers are of fixed speed ratio operates by assigning tasks to the server which will complete them soonest.

3. Distributing Work with Varying Server Speeds

When the arrival rate to a system like that shown in Figure 1 is so low that the controller C often can choose between completely idle servers, it will usually choose the fastest server since the work will be completed there the soonest. Only after queueing delays accumulate at the fastest server will the controller C wish to send work to a slower but idle server. As a result, the servers are not used equally. The fastest server performs the bulk of the work at very low arrival rates. If arrivals increase to approach the capacity of the system, the controller C is forced to distribute work in proportions equal to the speeds of the servers in order to find enough capacity to process the arrivals.

Given the arrival rate and server capacities, Buzen and Chen [3] have computed the probabilistic fractions of the arrival rate that should be sent to each of the servers for Poisson arrivals and service times to minimize average completion time.

3. Task Completion Time

An obvious approach to the problem posed in section 1 is to have the controller C route a new arrival to the server where it will be completed the soonest. This strategy is optimal in the case that all work arrives at the controller at time 0 [4].

In making this decision, the controller C must test each server by

adding the server's current backlog to the service time required for the new arrival at that server. The smallest total shows the server at which the new arrival will be completed the soonest.

Because the completion time of each arrival is being minimized, it is tempting to think that the average completion time for the system is minimized. This is not true!

This discrepancy between local optimization and global optimization was investigated by studying a complete transcript of arrival instants, routing decisions, and resultant completion times in a simulation. The effect of a routing decision was found to extend beyond the completion time of the work routed; it impacted subsequent or future arrivals. With high arrival rates the primary effect of the routing decision was on the completion time of subsequent arrivals.

4. The Future Effect

Imagine two servers. The faster completes work in 10 units of time. The slower one takes 20. Picture the situation using the time line or modified Gantt chart presented below. Arrivals are shown on the top line. The time spent in service is shown on the middle line. The completion times are summarized on the third line.

```

Arrivals:      1  2  3                      4      5
Server 1: 1111111111222222222233333333333344444444445555555555
Tot Time:           10          17          23          12          15

```

```

Arrivals:
Server 2: _____
Tot Time:
Avg Time: -

```

Average completion time: 15.4

Consider the effect on average completion time of placing arrival number 2 with the slower server. The situation would have been:

```

Arrivals:      1      3                      4      5
Server 1: 11111111113333333333_____44444444445555555555
Tot Time:           10          13          10          13
Avg Time: 11.5

```

```

Arrivals:      2
Server 2: 222222222222222222_____
Tot Time:                               20
Avg Time: 20.0

```

Average completion time: 13.2

The average completion time has been reduced significantly. In fact, even if the slower server were only one-third as fast, the overall average would have been improved.

A controller minimizing the completion time of each arrival would never have moved arrival number 2, however. Its completion time at the fast server is only 17...less than the 20 it experiences at the slower server.

The controller may, however, attempt to move arrival number 3 since its completion time at the fast server is 23, higher than an alternative server with completion time 20.

Arrivals:	1	2		4	5
Server 1:	11111111112222222222			44444444445555555555	
Tot Time:		10	17		10 13

Arrivals:		3
Server 2:	33333333333333333333	
Tot Time:		20

Average completion time: 14.0

Even though the pattern of busy time at the fastest server is identical to the previous case, moving arrival 3 does not reduce the average completion time as much as moving arrival 2.

It is also interesting to note that the average completion time is improved by allowing the fastest server to go idle. A policy which attempts to keep the fastest server busy does not result in lowest average completion time. This is usually true only in deadline scheduling and flowshop scheduling [5].

To optimize the assignment of the arrivals given in these examples, the total cost of an assignment on current and future completion times must be considered.

For example, the cost of assigning arrival 2 to server 1 is:

waiting time of 7
+ service time of 10
+ delay of 10 to arrival 3
+ delay of 2 to arrival 4
+ delay of 2 to arrival 5
= 31

Of this total of 31 time units, only 17 were seen by controller C as direct costs when arrival number 2 was routed. The other 14 were the effect of the decision on future arrivals. The total cost of assigning it to server 2 is only 20. This arrival pattern is optimized by moving arrival 2 to the slower server due to the "future effect".

A controller attempting to optimize its routing of arrivals must consider both the apparent cost of routing and the "future effect" cost in choosing a server. The apparent cost of routing is visible and easily determined. The "future effect" cost is more elusive.

5. Impact of the Future Effect

The magnitude of the future effect can be studied if a single server is considered in an M/G/1 environment. The expected cost of an extra arrival to an existing arrival pattern (its contribution to overall system delay) can be found analytically. It is the sum of the following:

1. The service time of the extra arrival. The service time is dependent on the speed of the server.

2. If the server is busy when the arrival occurs, the order of service can be permuted to consider the new arrival to preempt the server and go into service immediately [9]. The work in the remainder of the current busy period is delayed by the service time of the new arrival. The future effect cost is the service time of the new arrival multiplied by the number of tasks yet to complete in the original busy period which was interrupted.
3. If the server is busy when the arrival occurs, the existing busy period is extended by the length of the new service time and may bump into a subsequent busy period. All members of the subsequent busy period are delayed, but not by the entire amount of the new arrival's service time. The subsequent busy period, if delayed, may also move back and bump into another following busy period, and so on. The result is one much larger busy period composed of the smaller, individual busy periods which have been coalesced.
4. If the server is not busy when the arrival occurs, a succeeding arrival which begins a new busy period may be delayed if it arrives before service is completed to the extra arrival. Therefore, the recursive coalescing of busy periods discussed in the previous paragraph may occur.

The costs may be quantified as follows:

Let L = arrival rate to the server
 w = the backlog of work at the server
 b = the mean service time
 r = rho, the utilization of the server
 x = the service time of the new arrival

Then the cost of item 1 above is simply

$$x \quad (1)$$

The cost of item 2 above is the average remaining number of tasks in the existing busy period. Following Kleinrock's sub-busy period analysis [6] this is of size

$$\frac{w}{b(1-r)}$$

and multiplying by the cost (x) yields a total cost for this step of

$$\frac{x*w}{(1-r)*b} \quad (2)$$

It is not necessary to condition term (2) based on the probability of a busy server because it contributes nothing when the server is idle (since $w=0$ in that case).

Item 3 above occurs when the server is busy upon arrival, but item 4 is of the same size and occurs when the server is not busy upon arrival. The additional delay to subsequent busy periods may be calculated recursively. The expected cost to the first subsequent busy period is

$$F(x,L) = \text{Integral from } 0 \text{ to } x \text{ of } (x-z)e^{-Lz}L \, dz$$

Here z can be imagined to be the size of the existing gap between busy periods and x is the amount of time by which the first busy period

is extended. Factor $(x-z)$ is the delay to the following busy period, e^{-Lz} is the probability of no arrivals for period of time z , and L is the probability of arrival during interval dz .

The result is the average amount by which a subsequent busy period is moved back (if any). The second subsequent busy period will be coalesced only if $F(x,L)$ is larger than the idle time period preceding it. The average amount that the second subsequent busy period is delayed is $F(F(x,L),L)$. The third subsequent busy period is delayed on the average $F(F(F(x,L),L),L)$ and so on. Each busy period in the future has an average of $1/(1-r)$ customers ($M/G/1$). So the total cost due to items 3 and 4 is

$$\frac{F(x,L)}{(1-r)} + \frac{F(F(x,L),L)}{(1-r)} + \frac{F(F(F(x,L),L),L)}{(1-r)} + \dots \quad (3)$$

The total cost of an extra task is the sum of terms (1-3):

$$x + \frac{x*w}{b*(1-r)} + \frac{F(x,L)}{(1-r)} + \frac{F(F(x,L),L)}{(1-r)} + \frac{F(F(F(x,L),L),L)}{(1-r)} + \dots \quad (4)$$

Only term (2) depends on the size of the existing backlog at a server; it causes the total cost to rise linearly with existing backlog. Terms (1) and (3) define a fixed cost; term (1) is the known cost and term (3) is the future fixed cost.

6. Controllers Considering the Future Effect

A controller which takes the future effect into account in routing arrivals uses (4) to evaluate the additional cost of assigning new work to a particular server. Work is routed to the server where it will have the least additional total cost. Note that the directly observable costs (w and x) are taken into consideration but do not appear as terms by themselves.

While (4) is exact for an M/G/1 situation, the arrival pattern to a particular server for the situation pictured in Figure 1 will not have independent arrivals. After an arrival has been routed, a closely following arrival is not as likely to be routed to the same server. Nevertheless, the terms developed for M/G/1 can be used to approximate the total system impact. The results of the next section show that this approximation yields excellent results.

7. Evaluation

7.1. Method

A simulation experiment was carried out using SIMULA on a DECsystem-10 for 24,000 arrivals. Assignment to servers with speed ratios 4:2:1 was carried out by a controller which was programmable to test the different algorithms. The principal result of each simulation run was a system completion time averaged over all arrivals.

The controller algorithms simulated were:

1. Distribute probabilistically in proportion to server speeds.
2. Distribute probabilistically according to the Buzen-Chen fractions. (Used only for exponential service times.)
3. Send the work to the server which will complete it soonest.
4. Send the work to the server at which it will have the least total impact as judged by (4).
5. Send the work to the server at which the following simplified estimate of the impact is minimized:

$$x + \frac{w}{1-r}$$

This is approximately equivalent to terms (1) and (2).

6. Send the work to the server where it actually has the least total impact. This is accomplished by allowing an all-powerful observer to manipulate both past and future assignments until the overall system completion time is minimized.

All of the algorithms are feasible solutions to the problem posed in section 1 except for the last one which produces an ideal solution.

Simulations were conducted for arrival rates which represented .1 to .9 of the total system capacity in steps of .1. Fixed service times were used.

7.2. Results

The average time to completion is plotted against ρ , the fraction of system capacity represented by the arrival rate, in Figure 2. Each line is marked with a number representing the controller algorithm used for that set of simulation points.

The upper dotted line for algorithm "1" is the average time to completion when sending probabilistically proportional to server

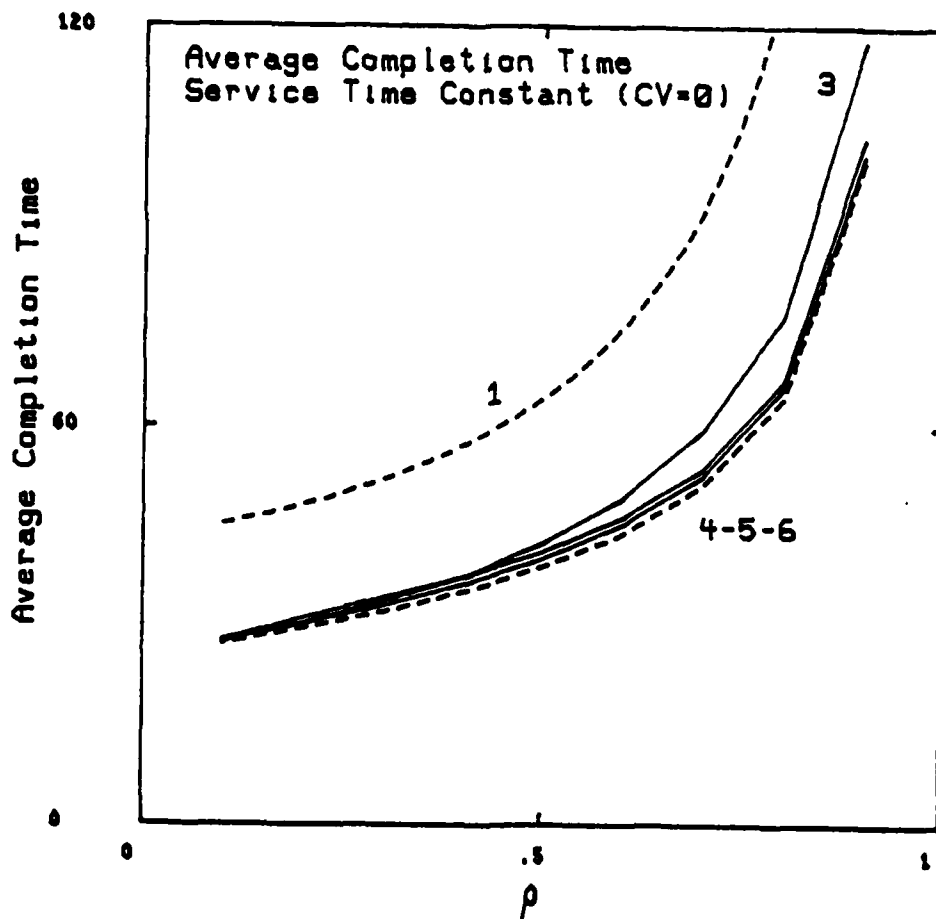


Figure 2

speed. It is markedly poor at all arrival rates because it does not use dynamic information on the state of backlogs to route work and will send new work to the slowest server even if the fastest one is idle.

The lower dotted line is the ideal average time to completion curve from algorithm 6.

The solid line for algorithm "3" shows the average time to completion when sending work to the server at which it will be completed soonest. At low arrival rates it performs well, but as the arrival rate rises its failure to take future arrivals into account results in higher than necessary average completion times.

Algorithms "4" and "5" take the future effect cost into account and have lower average completion times than algorithm "3". The choices made by "4" and "5" are so good that their delay curves lie very close to the ideal curve.

7.3. Discussion

Taking future costs into account reduces average completion time. The difference is more pronounced at higher arrival rates when the density of future arrivals is higher.

Controller algorithm 4 uses (4) to estimate the total additional increase in system completion time caused by routing the current arrival. To see how well (4) approximates the actual increase in completion times the simulation program was modified to record the extra impact on completion times due to each arrival. The data was collected

into buckets according to the value of w at the time the controller decision was made. The average increases in completion times for work routed to the fastest server are shown as triangles in Figure 3 at $\rho=0.7$. For comparison, the impact predicted by (4) is shown as a straight line. The relationship is good and shows that the "future effect" cost is about 3 times as large as the observable cost of a routing decision.

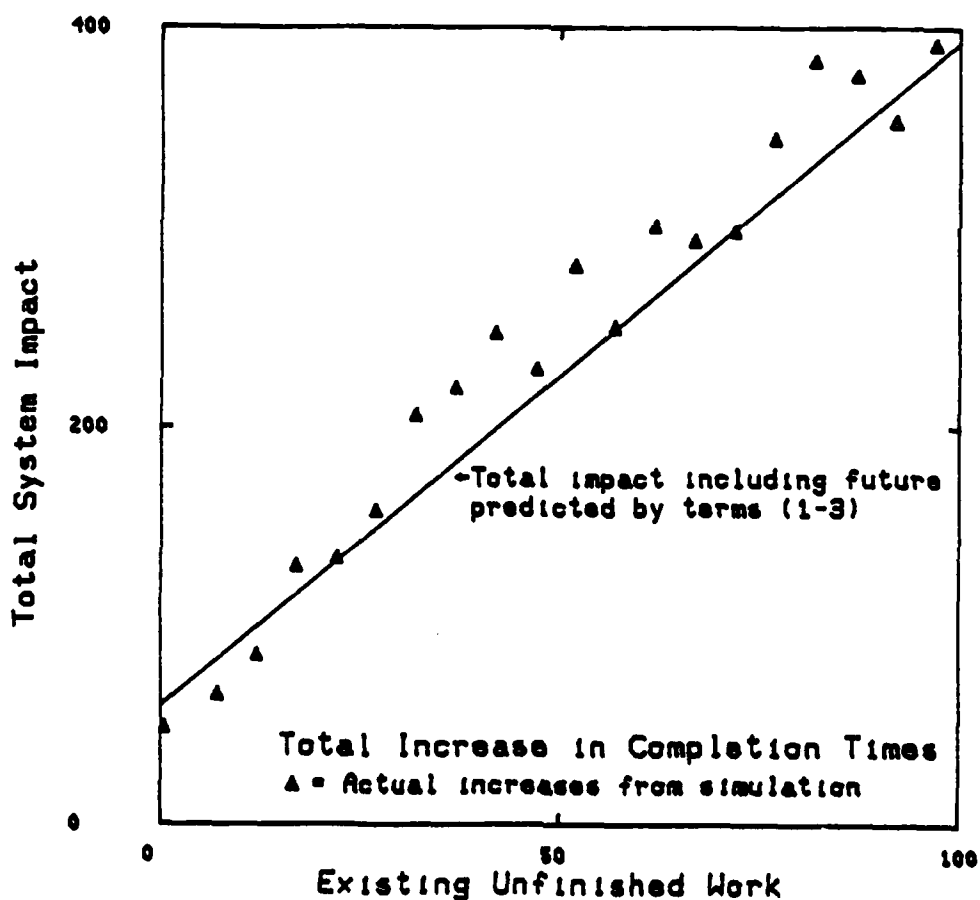


Figure 3

8. Limitations

Taking the future effect into account will not improve the average completion time under all circumstances.

8.1. Equal Speed Servers

In the case of equal speeds for the servers, the service time at all servers will be identical and terms (1) and (3) will yield identical values at all servers and may be disregarded. Therefore the least total effect will be produced by sending to the server which minimizes term (2) which can be considered a coefficient times the existing backlog. Sending to the server with lowest backlog is therefore the ideal strategy even considering the future effect in the case of equal speed servers.

8.2. Varying Service Times

If the service times are drawn from a distribution with considerable variance, the future effect is not as important as one having to do with the service time. In a loaded system the controller tends to equalize the sum of average service time, backlog, and future effect across all of the servers¹. If a new arrival has a much shorter than usual service requirement, it will usually be sent to a slower server. To see why this is so, consider the sum given above. When it is approximately equal across servers, the average service time is

¹ This sum is the result of (4). The algorithm sends to the server which has the smallest value. This procedure tends to equalize the sum across servers.

largest at the slowest server (by definition), and the backlog and future effect must be larger at the fastest server (since the total is about the same). As a result, a low service requirement arrival will tend to look better on the slower servers since the backlog and future effect are comparatively small and the service time is a small fraction of the total.

On the other hand, a large service requirement arrival usually is routed to the fastest server because the service time at a slower server would overshadow the larger backlog and future effect at the fastest server.

To check this quantitatively, the ideal distribution scheme was used to route arrivals with service times drawn from an exponential distribution ($\rho = .9$). The table below shows the fraction of work routed to each server (server 1 is the fastest, 3 the slowest).

Service Times	Fraction routed to server		
	1	2	3
Lowest 5%	.28	.35	.37
All	.50	.31	.19
Highest 5%	.72	.22	.06

The mean service times per server are correspondingly altered. The mean service time of work directed to the fastest server is 1.19 times the mean for all arrivals. The same factor for the medium speed server is .90 and for the slowest server it is only .67.

This tendency to distribute work according to service time further undermines the independent arrival assumption used by algorithm 4.

Algorithm 3 is sensitive to work distribution according to service time and is not substantially improved by considering the future costs. See Figure 4 which shows Algorithms 3, 4, 5, and 6 giving nearly identical performance. Algorithm 2 (Buzen-Chen fractions) is not as good as the others since it does not take current backlogs into account.

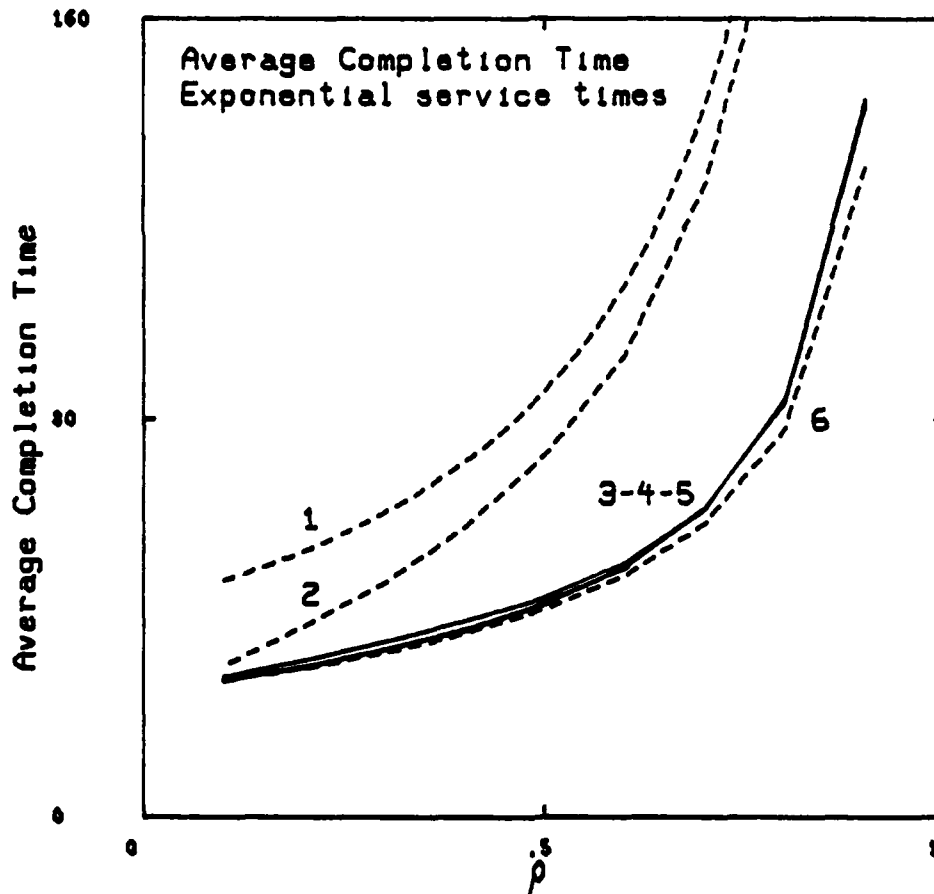


Figure 4

19. Weighting Unfinished Work

The additional cost of an arrival can be broken into terms (1) and (3) which are constant for a constant service time and arrival rate, and term (2) which depends upon the existing backlog. Therefore the cost including the future effect is $a_1x + a_2w$ where a_1 and a_2 are constants which depend upon the arrival rate and x is fixed for a given server. In algorithm 3 (send work to server with soonest completion time), the cost also takes this form but with $a_1=1$ and $a_2=1$. The future effect therefore modifies the relative weights to be given to the service time and unfinished work at a server.

These weights for algorithm 4 are:

rho	a_1	a_2	Ratio a_2/a_1
.1	1.054	1.111	1.054
.2	1.118	1.250	1.118
.3	1.198	1.429	1.192
.4	1.303	1.667	1.279
.5	1.448	2.000	1.381
.6	1.664	2.500	1.502
.7	2.023	3.333	1.648
.8	2.740	5.000	1.825
.9	4.889	10.000	2.046

The result of taking the future into account is to weight the amount of unfinished work more heavily than the service time. The exact amount of the extra weighting is not overly important. Algorithm 5 which sets a_1 to 1 and computes a_2 as in the table above gives average completion times almost indistinguishable from those of Algorithm 4.

20. Conclusion

In a work distribution model with servers of different speeds, the average completion time is not optimized by sending each task to the server at which it will be completed the soonest. The impact of any decision on future arrivals must be taken into account.

The information available to a controller may be categorized as static or dynamic, and "past" or "future". Static information is known by the controller and will not change. The static information includes the mean arrival rate and the server speeds. Dynamic information changes with time; dynamic information such as current backlogs must be exploited to achieve system goals. The "past" information is that body of knowledge of what has happened in the past. With Poisson arrivals, the complete state of past dynamic knowledge can be summarized for operational purposes by giving the current server backlogs. There is no "future" dynamic information...it is unknown. But there is an important piece of "future" static information: the mean arrival rate.

A successful controller must consider (a) The past dynamic information summarized by the current backlogs, (b) The current arrival and its requirements, and (c) The best projection of the future available from the "future" static information.

If no quantitative picture of the future is available, the future cost of a decision cannot be computed. But few arrivals are the "last" arrival and a controller can generally assume that the future arrival rate and pattern will be approximated by the recent past. This is

sufficient future information to allow a controller to estimate the probabilistic costs of its decisions on future arrivals.

A controller's best estimate of the future effect impact will in general be conditioned on all of the "past" information available. Even if the future effect has no simple relationship to this "past" information it may be estimated using historical information. Using some estimate will usually be better than no estimate, and the system is likely to be insensitive to all but gross errors.

REFERENCES

1. Agrawala, A. K., Tripathi, S., and Ricart, G., "Adaptive Routing Using a Virtual Waiting Time Technique", University of Maryland Technical Report TR- , November 1979.
2. Bruno, J., Coffman, E.G. Jr., and Sethi, R., Scheduling Tasks to Reduce Mean Finishing Time, Comm. ACM 17, 7, July 1974, pp. 382-387.
3. Buzen, J.P., and Chen, Peter P.S., Optimal Load Balancing in Memory Hierarchies, Information Processing 74, North Holland, pp. 271-275.
4. Clark, Douglas, Scheduling Independent Tasks on Non-Identical Parallel Machines to Minimize Mean Flow-Time, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa., June 1974.
5. Gonzalez, Mario J. Jr., Deterministic Processor Scheduling, Computing Surveys, Vol. 9, No. 3, Sep. 1977, pp. 173-203.
6. Kleinrock, Leonard, Queueing Systems. Vol I: Theory, John Wiley and Sons, Inc., New York, 1975, pp. 206-223.
7. Liu, Jane W.S., and Yang, Ai-Tsung, Optimal Scheduling of Independent Tasks on Heterogeneous Computing Systems, Proc. ACM National Conf. 1974, Vol 1, ACM, N.Y., 1974, pp. 38-45.
8. Ricart, Glenn, and Agrawala, A.K., Some Strategies for the Multiple Producer / Multiple Consumer Problem, Technical Report TR-816, University of Maryland Computer Science Department, October 1979.
9. Takács, L., Introduction to the Theory of Queues, Oxford University Press, New York, 1962